|  |  |
| --- | --- |
| Amadeus  Web Services  Implementation Guide |  |
| © 2005 – 2014 Amadeus IT Group SA. All Rights Reserved  Amadeus sas 485, Route du Pin Montard – Les Bouillides B.P. 69 06902 Sophia Antipolis CEDEX France  Amadeus Confidential Information Unauthorised use and disclosure strictly forbidden | Internet Booking Engine with Master Pricer  Version 1.2 (29 Dec 2015) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Document control | | | | |
| Security level | Restricted | | | |
| Company | Amadeus IT Group SA | | | |
| Department | Web Services Deployment and Governance | | | |
| Author | Sireethorn Limpanaset | | | |
| Reviewed by | Dylan White | | Date |  |
| Approved by |  | | Date |  |
| Version | Date | Change | Comment | By |
|  | 28 Oct 2014 | Document Creation | Draft version | Sireethorn L. |
| 1.1 | 27 Jan 2015 | Sanity check with Minirules |  | Mauricio Silva |
| 1.2 | 29 Dec 2015 | Update new version | Add FOP services | Sireethorn L. |

Index

1 Introduction 5

2 Prerequisites 5

2.1 Office ID Settings 5

2.2 LSS Permissions 5

2.3 Amadeus Web Services access 5

3 Concepts/Background 6

3.1 Internet Booking Engine 6

4 Functional Flow 6

4.1 List of Web Services 6

4.1.1 Search & Display 6

4.1.2 Select Recommendation 6

4.1.3 Passenger Detail + Book 7

4.1.4 MBO Operations 8

4.2 Flow Diagram 8

5 Search - Sample XML 10

5.1 Master Pricer Calendar Search (optional) 10

5.2 Master Pricer Travelboard Search 12

5.3 Tips 17

5.4 Error Handling 17

6 Select Recommendation - Sample XML 17

6.1 Fare\_InformativePricingWithoutPNR 18

6.2 Fare\_InformativeBestPricingWithoutPNR 23

6.3 Tips 28

6.4 Error Handling 29

7 Passenger Detail + Booking 30

7.1 Air\_SellFromRecommendation 30

7.2 PNR\_AddMultiElement 34

7.3 FOP\_CreateFormOfPayment / PNR\_AddMultiElement 36

7.4 Fare\_PricePNRWithBookingClass 38

7.5 Ticket\_CreateTSTFromPricing 39

7.6 PNR\_AddMultiElement 40

7.7 Tips 42

7.8 Error Handling 42

8 Certification Checklist 43

9 Certification Test Cases 46

10 References 48

# Introduction

This document describes the implementation process for an internet booking engine using Amadeus web services. It includes the steps for Search & Display, Select Recommendation and Passenger details + Book. It is intended for development team members such as business analysts, functional analysts and developers.

# Prerequisites

## Office ID Settings

An Office ID must be activated for Master Pricer products by the Amadeus Account Manager and access to the Amadeus Web Services must be granted by the Web Services Implementer.

Note: If the customer would like to implement a ticketing solution, the office ID must also be activated to operate as a ticketing office.

## LSS Permissions

A LSS user must be created under the customer’s office ID for Amadeus Web Services login. This is included in the setup of Amadeus Web Services access done by the Web Service Implementer.

## Amadeus Web Services access

Amadeus **W**eb **S**ervice **A**ccess **P**oint (WSAP)

* contains the list of services that can be accessed
* guarantees a secured access to the Amadeus applications
* defines a set of parameters to control the system behavior such as
  + the maximum number of concurrent sessions
  + the maximum volume of requests per second
  + time out values

The naming convention for WSAP

* WSAP: 1ASIWAAABBB
  + AAA: 3 letter code identifying the customer’s application.
  + BBB: 2 or 3 letter code identifying the customer in the Web Services data base

A WSAP is setup for TEST and PRODUCTION environments by the Web Service Implementer.

* The WSAP for TEST environment and the WSDL package are provided to the customer after the kick off call.
* The WSAP for PRODUCTION environment and the WSDL package are provided to the customer after the application is certified.

# Concepts/Background

The Internet Booking Engine (IBE) integrated with Master Pricer products (Master Pricer Calendar and Master Pricer TravelBoard) offers online users with a large choice of recommendations. It is ideal for cost-conscious travellers, focused on finding the cheapest fares for their journey. With the user-friendly interface and interactive business logic embedded in IBE, it allows the travellers to browse through the recommendations and ease the decision making process to make a reservation. As the result, the travellers can create a reservation with the flight and fare they prefer.

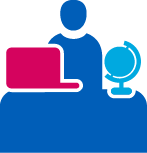
## Internet Booking Engine



**Amadeus**

**Web**

**Services**



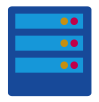
**Online user**

**Reply**

**Reply**

**IBE**

**Query**



**Query**

**Submit**

**Booking ref.**

Figure 1: Online users search and book a flight using IBE

# Functional Flow

The sample flow provided below is for the implementation of Web services for an Internet Booking Engine using Master Pricer.

## List of Web Services

Hereunder you will find a list of Amadeus Web Services that illustrates how to implement an internet booking engine using Master Pricer. These services are grouped into four different categories.

### Search & Display

**Fare\_MasterPricerCalendar (MPC)** is ideal for travellers and vacation planners who are more flexible with their travel dates and are willing to use this flexibility in order to obtain the cheapest fare. It is typically targeted at end-users more price sensitive than schedule sensitive.

**Fare\_MasterPricerTravelBoard** **(MPTB)** is ideal for cost-conscious travellers focused on finding the cheapest fares for specific travel dates.

### Select Recommendation

**Fare\_InformativePricingWithoutPNR** is used to price informatively an itinerary without the need to create a PNR. If a PNR already exists, it is neither taken into account nor updated. No pricing record (TST) is created to store the results.

**Fare\_InformativeBestPricingWithoutPNR** is used to perform informatively a ***Best Pricing*** request without the need to create a PNR. If a PNR already exists, it is neither taken into account nor updated. No pricing record (TST) is created to store the results.

**MiniRule\_GetFromPricing (FWR)** is used to display Amadeus Mini Rules which are a short and clear summary of the most important fare rules in a structured format for a given pricing. It is based on the rule restrictions that are effectively processed and validated for a given pricing context. This assures 100% up to date information.

**Fare\_CheckRules (FQN)** is used to display fare notes after Fare Display or after Pricing transactions. (This PSP service provides the same information as the cryptic FQN transaction.)

**Air\_FlightInfo (DO)** is used to display the latest known Flight Operational Information related to a specific flight/date. The Flight Information function provides Amadeus end-users with up-to-date information related to the operation of a particular flight before, during and after departure. Similarly, it provides irregularity and supplementary codes, which may be found in Availability, or Schedule displays to show a change in the flight operation.

**Air\_RetrieveSeatMap (ST)** works with the functionality of Amadeus Air to request seating information for a specific flight. A Seat Map contains valuable information pertaining to the cabin configuration, layout and seat status. This information, in most cases, is utilized to provide the end-user a preview of seat availability for advanced seat assignment purposes.

Note:  Air\_RetrieveSeatMap is only valid for airlines with an interactive seat map interface with Amadeus.

### Passenger Detail + Book

**Air\_SellFromRecommendation (SS)** is designed to sell a recommendation proposed by the **Low Fare Search** function. It has an optional selling algorithm, which is specified to overcome situations where a normal Sell would be rejected by the airlines.

**Fare\_PricePNRwithBookingClass (FXX)** is used to price itineraries. It can return one or several fare recommendations for the passenger(s) and for the itinerary of the active PNR. Only booking classes present in the flight segment of the PNR are considered for pricing.

**Fare\_PricePNRwithLowerFares (FXA)** is used to display the lowest available fare for a given itinerary. "Lowest available" means that this fare is applicable for a booking class if there are still enough seats available for all the passengers of the PNR. This class might not be the one currently present in the flight segment of the PNR, in this case rebooking might be necessary.

Note: After calling the Fare\_PricePNRWithBookingClass or Fare\_PricePNRwithLowerFaresfunction, the system keeps the recommendations stored internally for three minutes in a dedicated context. This context can be used to create a TST by using the Ticket\_CreateTSTFromPricing service.

**FOP\_CreateFormOfPayment** allows end user to perform the following creation operations: FP creation (associated or not to segments, chargeable SSR / SVC / MCO , passengers), Multiple Form of payment per FP line and Authorization process performed before FP creation.

**PNR\_AddMultiElement** allows the user to create an entire reservation in the Amadeus system with one transaction, bearing in mind that the full itinerary details must be known at the time of the function usage. This function allows many different elements to be combined into one transaction, thus being more efficient by reducing the number of transactions required from the client application.

**PNR\_Retrieve (RT)** is used to retrieve and display an active PNR or to redisplay a PNR during the current session.

**Ticket\_CreateTSTFromPricing** allows the Amadeus system to create a Transitional Stored Ticket (TST) after a confirmed pricing request such as **Fare\_PricePNRwithBookingClass.**

**Queue\_PlacePNR (QE)** is used to place a PNR onto one or more queues.

### MBO Operations

**DocIssuance\_IssueTicket (TTP)** allows the user to issue tickets against PNRs/TSTs.

## Flow Diagram

The following diagram shows the web services booking flow using the Master Pricer product.

There are four steps to create a flight reservation and issue tickets.

1. **Search & Display -** This section illustrates the low fare search process and recommendation proposal step by step.
2. **Select Recommendation** – This section shows the recommendation selection process including a basic price check before displaying it to the end user
3. **Passenger Details + Book** - This section provides a step by step sample of how to collect passenger details and check credit card validity prior to the creation of the Passenger Name Record (PNR) for a flight reservation
4. **MBO Operations** – This section displays the ticket issuance process using Amadeus Central Ticketing. This section is optional as ticketing can also be fulfilled using customer owned or 3rd party applications.

Figure 2: Shopping & Booking Business Flow

# Search - Sample XML

The search is the first step of the booking flow. The main purpose is to find the lowest and best available recommendations using our Low Fare Search products according to the criteria specified in the query, such as city pairs, date, cabin class, type of fares, passenger types and so on.

## Master Pricer Calendar Search (optional)

In order to perform a MPC search, certain mandatory parameters need to be included in the message.

* **A date of travel** - A date of travel must be specified for a MPC query
* **Origin/Destination** - At least one origin and one destination must be specified for a MPC query
* **Number of Passenger Seats** - The number of seats required for the traveling passengers must be specified. This number may not always be equal to the total number of passengers traveling. e.g., 2 Adults, 1 Child and 1 Infant - as an infant does not occupy a seat, this request requires only 3 seats for 4 traveling passengers
* **Associated Passenger Type Codes for Traveling Passengers -** Each traveling passenger must be associated to a passenger type code for a MPC search
* **Range of Dates** - The User has to specify for every MPC search a range of dates for each requested segment

Below is a sample MPC search query for a roundtrip journey including all mandatory elements

<Fare\_MasterPricerCalendar>

<numberOfUnit>

<unitNumberDetail>

<numberOfUnits>3</numberOfUnits>

Number of Passenger Seats

<typeOfUnit>PX</typeOfUnit>

</unitNumberDetail>

</numberOfUnit>

<paxReference>

Associated Passenger Type Codes for Traveling Passengers

**<ptc>ADT</ptc>**

<traveller>

<ref>1</ref>

</traveller>

<traveller>

<ref>2</ref>

</traveller>

</paxReference>

<paxReference>

**<ptc>CH</ptc>**

<traveller>

<ref>3</ref>

</traveller>

</paxReference>

<paxReference>

**<ptc>INF</ptc>**

<traveller>

<ref>1</ref>

<infantIndicator>1</infantIndicator>

</traveller>

</paxReference>

<fareOptions>

<pricingTickInfo>

<pricingTicketing>

<priceType>**RP**</priceType>

Fare options

<priceType>**RU**</priceType>

<priceType>**TAC**</priceType>

</pricingTicketing>

</pricingTickInfo>

</fareOptions>

<itinerary>

<requestedSegmentRef>

<segRef>1</segRef>

</requestedSegmentRef>

<departureLocalization>

<departurePoint>

<locationId>LON</locationId>

</departurePoint>

</departureLocalization>

Origin/Destination

<arrivalLocalization>

<arrivalPointDetails>

<locationId>TYO</locationId>

</arrivalPointDetails>

</arrivalLocalization>

<timeDetails>

<firstDateTimeDetail>

A date of travel

<date>030415</date>

</firstDateTimeDetail>

<rangeOfDate>

Range of Dates (-/+ 3 days)

<rangeQualifier>C</rangeQualifier>

<dayInterval>3</dayInterval>

</rangeOfDate>

</timeDetails>

</itinerary>

<itinerary>

<requestedSegmentRef>

<segRef>2</segRef>

</requestedSegmentRef>

<departureLocalization>

<departurePoint>

<locationId>TYO</locationId>

</departurePoint>

</departureLocalization>

<arrivalLocalization>

<arrivalPointDetails>

<locationId>LON</locationId>

</arrivalPointDetails>

</arrivalLocalization>

<timeDetails>

<firstDateTimeDetail>

<date>150415</date>

</firstDateTimeDetail>

<rangeOfDate>

<rangeQualifier>C</rangeQualifier>

<dayInterval>3</dayInterval>

</rangeOfDate>

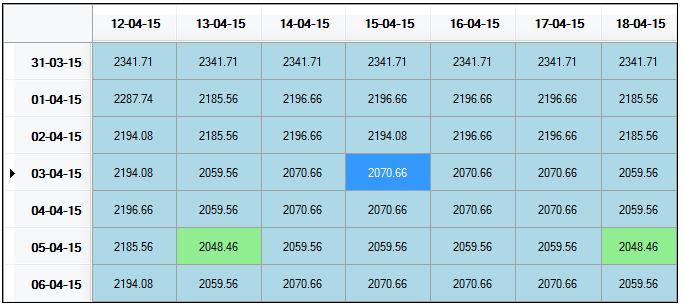
</timeDetails>

</itinerary>

</Fare\_MasterPricerCalendar>



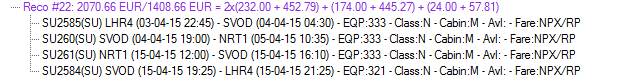
With the range of dates set to +/- 3days, MPC returns 49 recommendations, one for each combination of outbound and inbound dates.



On this example, the cheapest recommendations returned by MPC were priced at 2048.46 EUR and are highlighted on the matrix above in light green.

* 1. The outbound flight on 05-04-15 with the return flight on 13-04-15
  2. The outbound flight on 05-04-15 with the return flight on 18-04-15

After a recommendation of MPC is selected, a MPTB request should be sent as a follow-up transaction to get a list of lowest available recommendations (up to 250) for the selected travel dates to be presented to the end user.

The recommendation of MPC highlighted in blue on the matrix above, was selected for the follow-up MPTB request shown on the next chapter.

## Master Pricer Travelboard Search

A MPTB search is composed of the following mandatory elements:

* **A date of travel** - A date of travel must be specified for a MPTB query
* **Origin/Destination** - At least one origin and one destination must be specified for a MPTB query
* **Passenger Information** - the number of seats and the associated passenger type codes for the travelling passenger(s) are required. This number may not always be equal to the total number of passengers traveling. e.g., 2 Adults, 1 Child and 1 Infant - as an infant does not occupy a seat, this request requires only 3 seats for 4 traveling passengers
* **Number of Recommendations to be returned**

<Fare\_MasterPricerTravelBoardSearch>

<numberOfUnit>

<unitNumberDetail>

<numberOfUnits>3</numberOfUnits>

Number of Passenger Seats

<typeOfUnit>PX</typeOfUnit>

</unitNumberDetail>

<unitNumberDetail>

**RC** = number of recommendations.

The maximum number of recommendations in MPTB is **250**.

**<numberOfUnits>250</numberOfUnits>**

**<typeOfUnit>RC</typeOfUnit>**

</unitNumberDetail>

</numberOfUnit>

<paxReference>

Associated Passenger Type Codes for Traveling Passengers

**<ptc>ADT</ptc>**

<traveller>

<ref>1</ref>

</traveller>

<traveller>

<ref>2</ref>

</traveller>

</paxReference>

<paxReference>

**<ptc>CH</ptc>**

<traveller>

<ref>3</ref>

</traveller>

</paxReference>

<paxReference>

**<ptc>INF</ptc>**

<traveller>

<ref>1</ref>

<infantIndicator>1</infantIndicator>

</traveller>

</paxReference>

<fareOptions>

<pricingTickInfo>

<pricingTicketing>

<priceType>**RP**</priceType>

Fare options

<priceType>**RU**</priceType>

<priceType>**TAC**</priceType>

</pricingTicketing>

</pricingTickInfo>

</fareOptions>

<itinerary>

<requestedSegmentRef>

<segRef>1</segRef>

</requestedSegmentRef>

<departureLocalization>

<departurePoint>

<locationId>LON</locationId>

</departurePoint>

</departureLocalization>

Origin/Destination

<arrivalLocalization>

<arrivalPointDetails>

<locationId>TYO</locationId>

</arrivalPointDetails>

</arrivalLocalization>

<timeDetails>

<firstDateTimeDetail>

A date of travel

<date>030415</date>

</firstDateTimeDetail>

</timeDetails>

</itinerary>

<itinerary>

<requestedSegmentRef>

<segRef>2</segRef>

</requestedSegmentRef>

<departureLocalization>

<departurePoint>

<locationId>TYO</locationId>

</departurePoint>

</departureLocalization>

<arrivalLocalization>

<arrivalPointDetails>

<locationId>LON</locationId>

</arrivalPointDetails>

</arrivalLocalization>

<timeDetails>

<firstDateTimeDetail>

<date>150415</date>

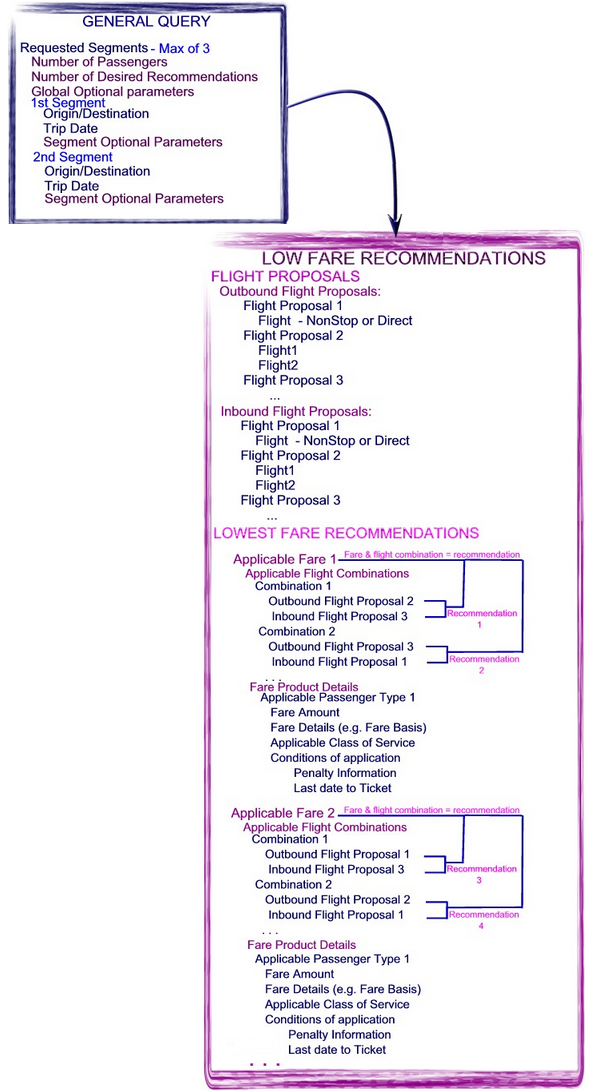
</firstDateTimeDetail>

</timeDetails>

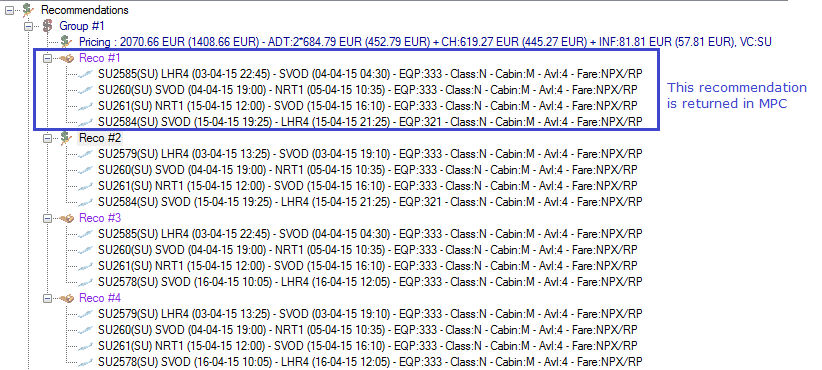
</itinerary>

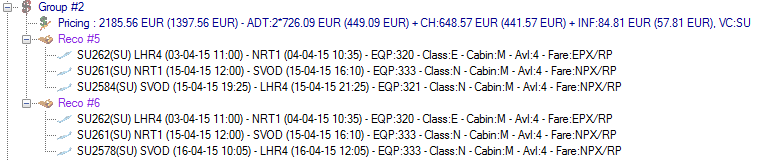
</Fare\_MasterPricerTravelBoardSearch>

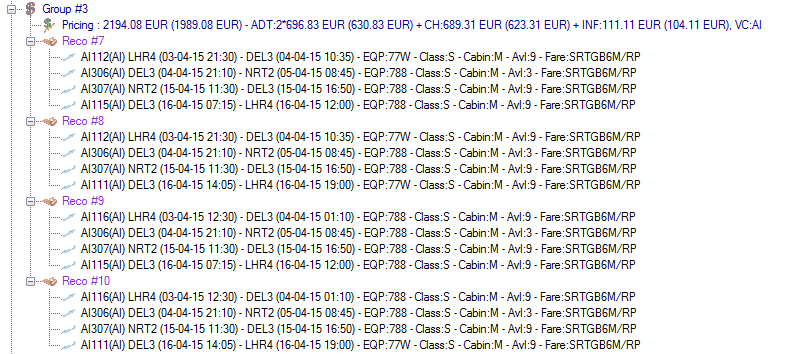




MPC returns the lowest available recommendation for each date combination within the range of dates requested. Once a recommendation isselected, a follow-up MPTB transaction can provide up to 250 recommendations, allowing a variety of choices with further airline diversity to be presented to the end user. Below is a sample subset of MPTB recommendations







## Tips

* Pooled / stateless session should be used for these flows
* A Master Pricer Calendar transaction should be followed-up by a Master Pricer TravelBoard transaction. The option “**TAC**” (Ticketability Pre-Check) should be specified in the MPC and in the MPTB fare option field. It enables to determine for each recommendation the list of validating carriers eligible for Ticketing. When a recommendation doesn’t have any eligible Validating carrier, then it is identified as non ticketable and it is not returned by Master Pricer
* If a Master Pricer Travel Board query is requested as a follow-up transaction of Master Pricer Calendar, the options in the Master Pricer Travel Board query must be the same options as in the Master Pricer Calendar query in order to target the same group of recommendations

## Error Handling

* The Master Pricer product must be activated at office ID level. If it is not activated, an error message will be returned as shown on the Master Pricer Calendar reply below.

<Fare\_MasterPricerCalendarReply>

<errorMessage>

<applicationError>

<applicationErrorDetail>

<error>**119**</error>

</applicationErrorDetail>

</applicationError>

<errorMessageText>

<freeTextQualification>

<textSubjectQualifier>**1**</textSubjectQualifier>

</freeTextQualification>

<description>User access denied for this transaction</description>

</errorMessageText>

</errorMessage>

</Fare\_MasterPricerCalendarReply>

* In the event that an error occurs, the application should record the sessionID, timestamp and query & reply for further investigation

# Select Recommendation - Sample XML

The Informative Pricing function can be used to verify pricing prior to end-user confirmation of the reservation.

Note: The sample queries below are created based on this recommendation

## Fare\_InformativePricingWithoutPNR

Certain information is mandatory on a Fare\_InformativePricingWithoutPNR query

* **Message Function** – messageFunction in messageDetails must have the value 741, standing for *Itinerary pricing for non-booked segments*.
* **Passenger information** 
  + **Passenger group ID** (segmentRepetitionControl/segmentControlDetails/quantity). Should be numbered 1, 2, 3...
  + **Number of passengers** **in this group** (segmentRepetitionControl/segmentControlDetails/numberOfUnits)
  + **Passenger IDs** (travellersID/travellerDetails/measurementValue). Contains the tattoos of the passengers. Should be numbered 1, 2, 3...
  + **Passenger Type Code (PTC), Fare Discounts (FD)** (ptcGroup/discountPTC/valueQualifier)
  + **Infant indicator** (please note that this is mandatory)
* **Segment information** 
  + **Airport code pair** (boardPointDetails/trueLocationId) and (offpointDetails/ trueLocationId)
  + **Departure Date** (flightDate/departureDate)
  + **Carrier code** (companyDetails/marketingCompany)
  + **Flight Number** (flightIdentification/flightNumber)
  + **Booking Class** (flightIdentification/bookingClass)

<Fare\_InformativePricingWithoutPNR>

<messageDetails>

<messageFunctionDetails>

Itinerary pricing for non-booked segments option

<businessFunction>1</businessFunction>

**<messageFunction>741</messageFunction>**

<responsibleAgency>1A</responsibleAgency>

</messageFunctionDetails>

</messageDetails>

**<corporateFares>**

**<corporateFareIdentifiers>**

Pricing option

**<fareQualifier>P</fareQualifier>**

**</corporateFareIdentifiers>**

**<corporateFareIdentifiers>**

**<fareQualifier>U</fareQualifier>**

**</corporateFareIdentifiers>**

**</corporateFares>**

**<passengersGroup>**

**<segmentRepetitionControl>**

**<segmentControlDetails>**

* Passenger group ID

Pax info

* No# of passengers
* Passenger IDs

**<quantity>1</quantity>**

**<numberOfUnits>2</numberOfUnits>**

**</segmentControlDetails>**

**</segmentRepetitionControl>**

**<travellersID>**

**<travellerDetails>**

**<measurementValue>1</measurementValue>**

**</travellerDetails>**

**<travellerDetails>**

**<measurementValue>2</measurementValue>**

**</travellerDetails>**

**</travellersID>**

**<ptcGroup>**

**<discountPtc>**

* Passenger Type codes

**<valueQualifier>ADT</valueQualifier>**

**</discountPtc>**

**</ptcGroup>**

**</passengersGroup>**

<passengersGroup>

<segmentRepetitionControl>

<segmentControlDetails>

<quantity>2</quantity>

<numberOfUnits>1</numberOfUnits>

</segmentControlDetails>

</segmentRepetitionControl>

<travellersID>

<travellerDetails>

<measurementValue>3</measurementValue>

</travellerDetails>

</travellersID>

<ptcGroup>

<discountPtc>

<valueQualifier>CH</valueQualifier>

</discountPtc>

</ptcGroup>

</passengersGroup>

<passengersGroup>

<segmentRepetitionControl>

<segmentControlDetails>

<quantity>3</quantity>

<numberOfUnits>1</numberOfUnits>

</segmentControlDetails>

</segmentRepetitionControl>

<travellersID>

<travellerDetails>

<measurementValue>1</measurementValue>

</travellerDetails>

</travellersID>

<ptcGroup>

Infant Association

<discountPtc>

<valueQualifier>INF</valueQualifier>

<fareDetails>

<qualifier>766</qualifier>

</fareDetails>

</discountPtc>

</ptcGroup>

</passengersGroup>

<pricingOptionsGroup>

**<pricingDetails>**

**<companyDetails>**

**Validating Carrier**

**<marketingCompany>SU</marketingCompany>**

**</companyDetails>**

**</pricingDetails>**

</pricingOptionsGroup>

<tripsGroup>

<originDestination>

<origin>TYO</origin>

<destination>LON</destination>

</originDestination>

<segmentGroup>

<segmentInformation>

<flightDate>

<departureDate>030415</departureDate>

</flightDate>

<boardPointDetails>

<trueLocationId>LHR</trueLocationId>

</boardPointDetails>

<offpointDetails>

<trueLocationId>NRT</trueLocationId>

</offpointDetails>

<companyDetails>

Segment information

<marketingCompany>SU</marketingCompany>

</companyDetails>

<flightIdentification>

<flightNumber>262</flightNumber>

<bookingClass>N</bookingClass>

</flightIdentification>

<flightTypeDetails>

* Connected segment info.

<flightIndicator>1</flightIndicator>

</flightTypeDetails>

<itemNumber>1</itemNumber>

</segmentInformation>

<trigger />

</segmentGroup>

<segmentGroup>

<segmentInformation>

<flightDate>

<departureDate>150415</departureDate>

</flightDate>

<boardPointDetails>

<trueLocationId>NRT</trueLocationId>

</boardPointDetails>

<offpointDetails>

<trueLocationId>SVO</trueLocationId>

</offpointDetails>

<companyDetails>

<marketingCompany>SU</marketingCompany>

</companyDetails>

<flightIdentification>

<flightNumber>261</flightNumber>

<bookingClass>N</bookingClass>

</flightIdentification>

<flightTypeDetails>

<flightIndicator>2</flightIndicator>

</flightTypeDetails>

<itemNumber>1</itemNumber>

</segmentInformation>

<trigger />

</segmentGroup>

<segmentGroup>

<segmentInformation>

<flightDate>

<departureDate>150415</departureDate>

</flightDate>

<boardPointDetails>

<trueLocationId>SVO</trueLocationId>

</boardPointDetails>

<offpointDetails>

<trueLocationId>LHR</trueLocationId>

</offpointDetails>

<companyDetails>

<marketingCompany>SU</marketingCompany>

</companyDetails>

<flightIdentification>

<flightNumber>2584</flightNumber>

<bookingClass>N</bookingClass>

</flightIdentification>

<flightTypeDetails>

<flightIndicator>2</flightIndicator>

</flightTypeDetails>

<itemNumber>2</itemNumber>

</segmentInformation>

<trigger />

</segmentGroup>

</tripsGroup>

</Fare\_InformativePricingWithoutPNR>

For this example, the number of seats on “N” class is not sufficient for 3 passengers.

/$AN03APRLHRNRT/ASU (1A)

\*\* AMADEUS AVAILABILITY - AN \*\* NRT NARITA INTL.JP 178 FR 03APR 0000

1 SU 262 J2 C2 D2 I2 Z4 Y4 B4 LHR 4 SVO D 1100 1650 E0/320

M4 U4 K4 H4 L4 Q4 T4 **E4 N1** R4 G0 V0

Therefore, Informative Pricing replies with an insufficient availability warning.

<textData>

<freeTextQualification>

<textSubjectQualifier>3</textSubjectQualifier>

<informationType>710</informationType>

</freeTextQualification>

<freeText>**WARNING - CLASS AVAILABILITY MAY NOT BE SUFFICIENT ON CERTAIN FLIGHTS**</freeText>

</textData>



## Fare\_InformativeBestPricingWithoutPNR

Certain information is mandatory in a Fare\_InformativeBestPricingWithoutPNR query

* **Message Function** in messageDetails –
  + **messageFunction** must have the value 741, standing for *Itinerary pricing for non-booked segments*.
  + ***additionalMessageFunction*** must have the value 170, standing for *Lowest fare across classes criteria*, in order to trigger a Best Pricing request.
* **Passenger information** 
  + **Passenger group ID** (segmentRepetitionControl/segmentControlDetails/quantity). Should be numbered 1, 2, 3...
  + **Number of passengers** **in this group** (segmentRepetitionControl/segmentControlDetails/numberOfUnits)
  + **Passenger IDs** (travellersID/travellerDetails/measurementValue). Contains the tattoos of the passengers. Should be numbered 1, 2, 3...
  + **Passenger Type Code (PTC), Fare Discounts (FD)** (ptcGroup/discountPTC/valueQualifier)
  + **Infant indicator** (please note that this is mandatory)
* **Segment information** 
  + **Airport code pair**: boardPointDetails/trueLocationId and offpointDetails/trueLocationId
  + **Departure Date** (flightDate/departureDate)
  + **Carrier code** (companyDetails/marketingCompany)
  + **Flight Number** (flightIdentification/flightNumber)
  + **Booking Class** (flightIdentification/bookingClass)

<Fare\_InformativeBestPricingWithoutPNR>

Pax info

* No# of passengers
* Passenger IDs
* Passenger Type codes

<messageDetails>

<messageFunctionDetails>

Itinerary pricing for non-booked segments option

<businessFunction>1</businessFunction>

**<messageFunction>741</messageFunction>**

<responsibleAgency>1A</responsibleAgency>

**<additionalMessageFunction>170</additionalMessageFunction>**

</messageFunctionDetails>

</messageDetails>

**<corporateFares>**

**<corporateFareIdentifiers>**

**<fareQualifier>P</fareQualifier>**

**</corporateFareIdentifiers>**

Pricing option

**<corporateFareIdentifiers>**

**<fareQualifier>U</fareQualifier>**

**</corporateFareIdentifiers>**

**</corporateFareInfo>**

**<passengersGroup>**

**<segmentRepetitionControl>**

**<segmentControlDetails>**

* Passenger group ID

**<quantity>1</quantity>**

**<numberOfUnits>2</numberOfUnits>**

**</segmentControlDetails>**

**</segmentRepetitionControl>**

**<travellersID>**

**<travellerDetails>**

**<measurementValue>1</measurementValue>**

**</travellerDetails>**

**<travellerDetails>**

**<measurementValue>2</measurementValue>**

**</travellerDetails>**

**</travellersID>**

**<ptcGroup>**

**<discountPtc>**

**<valueQualifier>ADT</valueQualifier>**

**</discountPtc>**

**</ptcGroup>**

**</passengersGroup>**

<passengersGroup>

<segmentRepetitionControl>

<segmentControlDetails>

<quantity>2</quantity>

<numberOfUnits>1</numberOfUnits>

</segmentControlDetails>

</segmentRepetitionControl>

<travellersID>

<travellerDetails>

<measurementValue>3</measurementValue>

</travellerDetails>

</travellersID>

<ptcGroup>

<discountPtc>

<valueQualifier>CH</valueQualifier>

</discountPtc>

</ptcGroup>

</passengersGroup>

<passengersGroup>

<segmentRepetitionControl>

<segmentControlDetails>

<quantity>3</quantity>

<numberOfUnits>1</numberOfUnits>

</segmentControlDetails>

</segmentRepetitionControl>

<travellersID>

Infant Association

Segment information

<travellerDetails>

<measurementValue>1</measurementValue>

</travellerDetails>

</travellersID>

<ptcGroup>

<discountPtc>

<valueQualifier>INF</valueQualifier>

<fareDetails>

<qualifier>766</qualifier>

</fareDetails>

</discountPtc>

</ptcGroup>

</passengersGroup>

<pricingOptionsGroup>

<pricingDetails>

<priceTicketDetails>

Verify Original Availability option

<indicators>VOA</indicators>

</priceTicketDetails>

<companyDetails>

Validating Carrier

<marketingCompany>SU</marketingCompany>

</companyDetails>

</pricingDetails>

</pricingOptionsGroup>

<tripsGroup>

<originDestination>

<origin>TYO</origin>

<destination>LON</destination>

</originDestination>

<segmentGroup>

<segmentInformation>

<flightDate>

<departureDate>030415</departureDate>

</flightDate>

<boardPointDetails>

<trueLocationId>LHR</trueLocationId>

</boardPointDetails>

<offpointDetails>

<trueLocationId>NRT</trueLocationId>

</offpointDetails>

<companyDetails>

<marketingCompany>SU</marketingCompany>

</companyDetails>

<flightIdentification>

<flightNumber>262</flightNumber>

<bookingClass>N</bookingClass>

</flightIdentification>

<flightTypeDetails>

* Connected segment info.

<flightIndicator>1</flightIndicator>

</flightTypeDetails>

<itemNumber>1</itemNumber>

</segmentInformation>

</segmentGroup>

<segmentGroup>

<segmentInformation>

<flightDate>

<departureDate>150415</departureDate>

</flightDate>

<boardPointDetails>

<trueLocationId>NRT</trueLocationId>

</boardPointDetails>

<offpointDetails>

<trueLocationId>SVO</trueLocationId>

</offpointDetails>

<companyDetails>

<marketingCompany>SU</marketingCompany>

</companyDetails>

<flightIdentification>

<flightNumber>261</flightNumber>

<bookingClass>N</bookingClass>

</flightIdentification>

<flightTypeDetails>

<flightIndicator>2</flightIndicator>

</flightTypeDetails>

<itemNumber>1</itemNumber>

</segmentInformation>

</segmentGroup>

<segmentGroup>

<segmentInformation>

<flightDate>

<departureDate>150415</departureDate>

</flightDate>

<boardPointDetails>

<trueLocationId>SVO</trueLocationId>

</boardPointDetails>

<offpointDetails>

<trueLocationId>LHR</trueLocationId>

</offpointDetails>

<companyDetails>

<marketingCompany>SU</marketingCompany>

</companyDetails>

<flightIdentification>

<flightNumber>2584</flightNumber>

<bookingClass>N</bookingClass>

</flightIdentification>

<flightTypeDetails>

<flightIndicator>2</flightIndicator>

</flightTypeDetails>

<itemNumber>2</itemNumber>

</segmentInformation>

</segmentGroup>

</tripsGroup>

</Fare\_InformativeBestPricingWithoutPNR>

For this example, the booking class “N” specified in the query is not available. With the option VOA (Verify Original Availability), Informative Best Pricer returns the lowest available recommendation instead with booking class “E”.

/$AN03APRLHRNRT/ASU (1A)

\*\* AMADEUS AVAILABILITY - AN \*\* NRT NARITA INTL.JP 178 FR 03APR 0000

1 SU 262 J2 C2 D2 I2 Z4 Y4 B4 LHR 4 SVO D 1100 1650 E0/320

M4 U4 K4 H4 L4 Q4 T4 **E4 N0** R4 G0 V0

SVO D NRT 1 1900 1035+1E0/333 15:35

<Fare\_InformativeBestPricingWithoutPNRReply>

..

<segmentLevelGroup>

<segmentInformation>

<flightDate>

<departureDate>030415</departureDate>

</flightDate>

<boardPointDetails>

<trueLocationId>LHR</trueLocationId>

</boardPointDetails>

<offpointDetails>

<trueLocationId>NRT</trueLocationId>

</offpointDetails>

<companyDetails>

<marketingCompany>SU</marketingCompany>

</companyDetails>

<flightIdentification>

<flightNumber>262</flightNumber>

**<bookingClass>E</bookingClass>**

<operationalSuffix>X</operationalSuffix>

</flightIdentification>

<itemNumber>1</itemNumber>

</segmentInformation>

..

<segmentInformation>

<flightDate>

<departureDate>150415</departureDate>

</flightDate>

<boardPointDetails>

<trueLocationId>NRT</trueLocationId>

</boardPointDetails>

<offpointDetails>

<trueLocationId>SVO</trueLocationId>

</offpointDetails>

<companyDetails>

<marketingCompany>SU</marketingCompany>

</companyDetails>

<flightIdentification>

<flightNumber>261</flightNumber>

**<bookingClass>E</bookingClass>**

<operationalSuffix>X</operationalSuffix>

</flightIdentification>

<flightTypeDetails>

<flightIndicator>T</flightIndicator>

</flightTypeDetails>

<itemNumber>2</itemNumber>

</segmentInformation>

..

<segmentInformation>

<flightDate>

<departureDate>150415</departureDate>

</flightDate>

<boardPointDetails>

<trueLocationId>SVO</trueLocationId>

</boardPointDetails>

<offpointDetails>

<trueLocationId>LHR</trueLocationId>

</offpointDetails>

<companyDetails>

<marketingCompany>SU</marketingCompany>

</companyDetails>

<flightIdentification>

<flightNumber>2584</flightNumber>

**<bookingClass>E</bookingClass>**

<operationalSuffix>X</operationalSuffix>

</flightIdentification>

<itemNumber>3</itemNumber>

</segmentInformation>

</segmentLevelGroup>

..

</Fare\_InformativeBestPricingWithoutPNRReply>



## Tips

* To avoid getting an incorrect pricing, connecting flights should be specified accordingly in the Informative pricing request. In this example the first segment (outbound flight) is independent. The second and third (inbound flights) are connected. Two bounds are defined (flightIndicator 1 and 2)



For the outbound flight -> SU262, flightIndicator= 1 and itemNumber = 1

For the inbound flight -> SU261, flightIndicator= 2 and itemNumber = 1

-> SU2584, flightIndicator= 2 and itemNumber = 2

* To avoid fare discrepancies between pricing and Master Pricer replies, the Validating Carrier provided by Master Pricer should be applied in the validating carrier option of the pricing query. The recommended Validating Carriers in the MPTB reply can be found at

Fare\_MasterPricerTravelBoardSearchReply/recommendation/paxFareProduct/paxFareDetail

<paxFareDetail>

<paxFareNum>1</paxFareNum>

<totalFareAmount>684.79</totalFareAmount>

<totalTaxAmount>452.79</totalTaxAmount>

<codeShareDetails>

**<transportStageQualifier>V</transportStageQualifier>**

**<company>SU</company>**

</codeShareDetails>

</paxFareDetail>

* By default, the system assumes that the class indicated in the pricing request is available. if this is not the case, the service may return an insufficient availability warning.

<textData>

<freeTextQualification>

<textSubjectQualifier>3</textSubjectQualifier>

<informationType>710</informationType>

</freeTextQualification>

<freeText>**WARNING - CLASS AVAILABILITY MAY NOT BE SUFFICIENT ON CERTAIN FLIGHTS**</freeText>

</textData>

* In order to properly process the availability of the class of booking being sent in the Informative Best Pricing request, the “Check Availability” option must be used indicating if the class is to be assumed available or if the process should check it’s availability.
  + If the value of this option is set to **VOA**, no assumption on the availability is made and the process checks if the class sent in the request is available when searching for the lowest available fare
  + If the value is set to **NVO**, the system assumes that the class indicated in the pricing request is available without any check. So it may return a class that is not available if the cheapest recommendation is for the booking class provided in the query and this booking class is sold out. On this case, no warning is sent.

Note: Infomative Best Pricer version 13.1 and above checks the availability of the class sent in the request by default. If needed, the NVO option can be sent to inhibit this check.

* The MiniRule\_GetFromPricing service and/or the Fare\_CheckRules service can be sent after Fare\_InformativePricingWithoutPNR or Fare\_InformativeBestPricingWithoutPNR to get the Fare rule content.

## Error Handling

* At this step, if Fare\_InformativePricingWithoutPNR returns the error “No Fare found” or an insufficient availability warning, Fare\_InformativeBestPricingWithoutPNR should be called to get the new alternative recommendation. Fare\_InformativeBestPricingWithoutPNR always returns the same flights requested, but the booking class may be changed depending on the availability of these flights
* When the additionalMessageFunction = 170 (Lowest fare across classes criteria) is not specified, the error below will be returned.

<Fare\_InformativeBestPricingWithoutPNR>

<messageDetails>

<messageFunctionDetails>

<businessFunction>1</businessFunction>

<messageFunction>741</messageFunction>

<responsibleAgency>1A</responsibleAgency>

</messageFunctionDetails>

</messageDetails>

…

..

</Fare\_InformativeBestPricingWithoutPNR>

<Fare\_InformativeBestPricingWithoutPNRReply>

<messageDetails>

<messageFunctionDetails>

<businessFunction>1</businessFunction>

<messageFunction>741</messageFunction>

<responsibleAgency>1A</responsibleAgency>

<additionalMessageFunction>170</additionalMessageFunction>

</messageFunctionDetails>

<responseType>8</responseType>

</messageDetails>

<errorGroup>

<errorOrWarningCodeDetails>

<errorDetails>

<errorCode>477</errorCode>

<errorCategory>EC</errorCategory>

</errorDetails>

</errorOrWarningCodeDetails>

<errorWarningDescription>

<freeTextDetails>

<textSubjectQualifier>3</textSubjectQualifier>

<source>M</source>

<encoding>1</encoding>

</freeTextDetails>

<freeText>**Invalid message details. Be sure to ask for a Non Integrated Pricing**</freeText>

</errorWarningDescription>

</errorGroup>

</Fare\_InformativeBestPricingWithoutPNRReply>

* In the event that an error occurs, the application should record the sessionID, timestamp and query & reply for further investigation

# Passenger Detail + Booking

When creating bookings, all five mandatory elements (Name element (NM), Itinerary Element, Received from (RF), Contact element (AP) and Ticket Element (TK) need to be present before committing the PNR. Also, the PNR must be priced and the TST must be created prior to sending it for ticketing.

The standard steps for creating a booking

1. Air\_SellFromRecommendation to add Itinerary Elements
2. PNR\_AddMulitElement to add Name element (NM), Itinerary Element, Received from (RF), Contact element (AP), Ticket Element (TK)
3. Fare\_PricePNRWithBookingClass(FXX) to price itineraries
4. Ticket\_CreateTSTFromPricing to store the TST from the pricing
5. PNR\_AddMulitElement to commit the PNR

After committing the PNR, an Amadeus record locator will be created and you can use it as the reference for your booking.

## Air\_SellFromRecommendation

Air\_SellFromRecommendation is a basic low fare sell service composed of the following minimum mandatory elements:

* **Message Function** in messageFunctionDetails
  + **messageFunction** must have the value 183, standing for *Lowest fare across airline, flight, class criteria*
  + **additionalMessageFunction** – there are two Sell Optimization Algorithms which can overcome situations where a normal Sell would be rejected by the airlines.
    - **M1** = Trigger Sell Optimization Algorithm, option cancel all if unsuccessful
    - **M2** = Trigger Sell Optimization Algorithm, option keep all confirmed if unsuccessful.
* **Itinerary** is composed of
  + Origin and Destination: departure and arrival cities
  + A list of segments, composed of
    - flight number, board and off airports, departure date
    - number of passengers requested

<Air\_SellFromRecommendation>

<messageActionDetails>

<messageFunctionDetails>

Lowest Fare Search Message

<messageFunction>183</messageFunction>

Algorithm

<additionalMessageFunction>M1</additionalMessageFunction>

</messageFunctionDetails>

</messageActionDetails>

<itineraryDetails>

<originDestinationDetails>

Origin and Destination

**<origin>LHR</origin>**

**<destination>NRT</destination>**

</originDestinationDetails>

<message>

<messageFunctionDetails>

Lowest Fare Search Message

<messageFunction>183</messageFunction>

</messageFunctionDetails>

</message>

<segmentInformation>

<travelProductInformation>

<flightDate>

<departureDate>030415</departureDate>

</flightDate>

<boardPointDetails>

<trueLocationId>LHR</trueLocationId>

</boardPointDetails>

<offpointDetails>

Segment Details

<trueLocationId>NRT</trueLocationId>

</offpointDetails>

<companyDetails>

<marketingCompany>SU</marketingCompany>

</companyDetails>

<flightIdentification>

<flightNumber>262</flightNumber>

<bookingClass>E</bookingClass>

</flightIdentification>

</travelProductInformation>

<relatedproductInformation>

Number of passengers requested

<quantity>3</quantity>

<statusCode>NN</statusCode>

</relatedproductInformation>

</segmentInformation>

</itineraryDetails>

<itineraryDetails>

<originDestinationDetails>

**<origin>NRT</origin>**

**<destination>LHR</destination>**

</originDestinationDetails>

<message>

<messageFunctionDetails>

<messageFunction>183</messageFunction>

</messageFunctionDetails>

</message>

<segmentInformation>

<travelProductInformation>

<flightDate>

<departureDate>150415</departureDate>

</flightDate>

<boardPointDetails>

<trueLocationId>NRT</trueLocationId>

</boardPointDetails>

<offpointDetails>

<trueLocationId>SVO</trueLocationId>

</offpointDetails>

<companyDetails>

<marketingCompany>SU</marketingCompany>

</companyDetails>

<flightIdentification>

<flightNumber>261</flightNumber>

<bookingClass>E</bookingClass>

</flightIdentification>

</travelProductInformation>

<relatedproductInformation>

<quantity>3</quantity>

<statusCode>NN</statusCode>

</relatedproductInformation>

</segmentInformation>

<segmentInformation>

<travelProductInformation>

<flightDate>

<departureDate>150415</departureDate>

</flightDate>

<boardPointDetails>

<trueLocationId>SVO</trueLocationId>

</boardPointDetails>

<offpointDetails>

<trueLocationId>LHR</trueLocationId>

</offpointDetails>

<companyDetails>

<marketingCompany>SU</marketingCompany>

</companyDetails>

<flightIdentification>

<flightNumber>2584</flightNumber>

<bookingClass>E</bookingClass>

</flightIdentification>

</travelProductInformation>

<relatedproductInformation>

<quantity>3</quantity>

<statusCode>NN</statusCode>

</relatedproductInformation>

</segmentInformation>

</itineraryDetails>

</Air\_SellFromRecommendation>



This is the result after sending the request above.

/$RP/NCE1A0950/

1 SU 262 E 03APR 5 LHRNRT DK3 1100 1035 04APR E 1 EQV MM

2 SU 261 E 15APR 3 NRTSVO DK3 1200 1610 15APR E 0 333 LS

SEE RTSVC

3 SU2584 E 15APR 3 SVOLHR DK3 1925 2125 15APR E 0 321 L

SEE RTSVC

According to the Fare\_InformativeBestPricingWithoutPNR result, it shows that the booking class “N” is not available, and “E” is proposed instead to be the lowest available recommendation.

With Algorithm “M1” if the booking class “N” is sold, the result of Air\_SellFromRecommendation will be returned as below.

For the outbound (1st O&D: LHR ->NRT)

* SU262(LHR–NRT) The booking class “N” cannot be sold, the status “UNS” – Unserviceable is returned since the class is not available

For the inbound (2nd O&D: NRT -> LHR)

* SU262 (NRT-SVO) and SU2584 (SVO-LHR) will be returned with status as “RQ” - Requested which are not sold

No segment is sold with **M1** when one segment is unable to sell.



Confirmed and/or waitlisted segments (when the waitlist option is set) remain sold with **M2** when a segment is unable to sell.

Recommendations

Group #1

Pricing : 13832.50 ARS (7636.50 ARS) - ADT:13832.50 ARS (7636.50 ARS), VC:YO

Reco #1

SU262(SU) LHR4 (03-04-15 11:00) - NRT1 (04-04-15 10:35) - EQP:320 - Class:N - Cabin:M - Avl:5 - Fare:NPX/RP => UNS

SU261(SU) NRT1 (15-04-15 12:00) - SVOD (15-04-15 16:10) - EQP:333 - Class:N - Cabin:M - Avl:7 - Fare:NPX/RP => OK

SU2584(SU) SVOD (15-04-15 19:25) - LHR4 (15-04-15 21:25) - EQP:321 - Class:N - Cabin:M - Avl:7 - Fare:NPX/RP => OK

## PNR\_AddMultiElement

To add the mandatory elements; Name element (NM), Itinerary Element, Received from (RF), Contact element (AP), Ticket Element (TK).

<PNR\_AddMultiElements>

<pnrActions>

<optionCode>0</optionCode>

</pnrActions>

Passenger information

**<travellerInfo>**

**<elementManagementPassenger>**

**<reference>**

**<qualifier>PR</qualifier>**

**<number>1</number>**

**</reference>**

**<segmentName>NM</segmentName>**

**</elementManagementPassenger>**

**<passengerData>**

**<travellerInformation>**

**<traveller>**

1st Adult associated with infant

**<surname>FamilyName</surname>**

**<quantity>2</quantity>**

**</traveller>**

**<passenger>**

**<firstName>AdultNameOne</firstName>**

**<type>ADT</type>**

**<infantIndicator>2</infantIndicator>**

**</passenger>**

**<passenger>**

**<firstName>BabyNameOne</firstName>**

**<type>INF</type>**

**</passenger>**

**</travellerInformation>**

**</passengerData>**

**</travellerInfo>**

**<travellerInfo>**

**<elementManagementPassenger>**

**<reference>**

**<qualifier>PR</qualifier>**

**<number>2</number>**

**</reference>**

**<segmentName>NM</segmentName>**

**</elementManagementPassenger>**

**<passengerData>**

**<travellerInformation>**

2nd Adult

**<traveller>**

**<surname>FamilyName</surname>**

**<quantity>1</quantity>**

**</traveller>**

**<passenger>**

**<firstName>AdultNameTwo</firstName>**

**<type>ADT</type>**

**</passenger>**

**</travellerInformation>**

**</passengerData>**

**</travellerInfo>**

**<travellerInfo>**

**<elementManagementPassenger>**

**<reference>**

**<qualifier>PR</qualifier>**

**<number>3</number>**

**</reference>**

**<segmentName>NM</segmentName>**

**</elementManagementPassenger>**

**<passengerData>**

**<travellerInformation>**

**<traveller>**

1st Child

**<surname>FamilyName</surname>**

**<quantity>1</quantity>**

**</traveller>**

**<passenger>**

**<firstName>ChildNameOne</firstName>**

**<type>CHD</type>**

**</passenger>**

**</travellerInformation>**

**</passengerData>**

**</travellerInfo>**

<dataElementsMaster>

<marker1 />

<dataElementsIndiv>

<elementManagementData>

<segmentName>AP</segmentName>

</elementManagementData>

<freetextData>

<freetextDetail>

AP element

<subjectQualifier>3</subjectQualifier>

<type>P21</type>

</freetextDetail>

<longFreetext></longFreetext>

</freetextData>

</dataElementsIndiv>

**<dataElementsIndiv>**

**<elementManagementData>**

**<segmentName>TK</segmentName>**

**</elementManagementData>**

**<ticketElement>**

TK element

**<ticket>**

**<indicator>OK</indicator>**

**</ticket>**

**</ticketElement>**

**</dataElementsIndiv>**

**<dataElementsIndiv>**

**<elementManagementData>**

**<segmentName>RF</segmentName>**

**</elementManagementData>**

**<freetextData>**

RF element

**<freetextDetail>**

**<subjectQualifier>3</subjectQualifier>**

**<type>P22</type>**

**</freetextDetail>**

**<longFreetext>AWSUI</longFreetext>**

**</freetextData>**

**</dataElementsIndiv>**

</dataElementsMaster>

</PNR\_AddMultiElements>



After sending the request above, the mandatory elements are added to the booking.

/$RP/NCE1A0950/

**RF AWSUI**

**1.FAMILYNAME/ADULTNAMEONE(ADT)(INF/BABYNAMEONE)**

**2.FAMILYNAME/ADULTNAMETWO(ADT)**

**3.FAMILYNAME/CHILDNAMEONE(CHD)**

4 SU 262 E 03APR 5 LHRNRT DK3 1100 1035 04APR E 1 EQV MM

5 SU 261 E 15APR 3 NRTSVO DK3 1200 1610 15APR E 0 333 LS

SEE RTSVC

6 SU2584 E 15APR 3 SVOLHR DK3 1925 2125 15APR E 0 321 L

SEE RTSVC

**7 AP NCE 33492943273 - AMADEUS PRODUCT PLANNING - A**

**8 TK OK07OCT/NCE1A0950**

9 SSR INFT SU NN1 FAMILYNAME/BABYNAMEONE 07OCT14/S4/P1

10 SSR INFT SU NN1 FAMILYNAME/BABYNAMEONE 07OCT14/S5/P1

11 SSR INFT SU NN1 FAMILYNAME/BABYNAMEONE 07OCT14/S6/P1

12 SSR CHLD SU HK1/P3

## FOP\_CreateFormOfPayment / PNR\_AddMultiElement

To add the FP elements, either using FOP\_CreateFormOfPayment or PNR\_AddMultiElement

<FOP\_CreateFormOfPayment>

<fopGroup>

<fopReference/>

<mopDescription>

<fopSequenceNumber>

<sequenceDetails>

<number>1</number>

</sequenceDetails>

</fopSequenceNumber>

<mopDetails>

<fopPNRDetails>

<fopDetails>

<fopCode>CCVI</fopCode>

</fopDetails>

</fopPNRDetails>

</mopDetails>

<paymentModule>

<groupUsage>

<attributeDetails>

<attributeType>FP</attributeType>

</attributeDetails>

</groupUsage>

<mopInformation>

<fopInformation>

<formOfPayment>

<type>CC</type>

</formOfPayment>

</fopInformation>

<dummy/>

<creditCardData>

<creditCardDetails>

<ccInfo>

<vendorCode>VI</vendorCode>

<cardNumber>4444333322221111</cardNumber>

<securityId>**123**</securityId>

<expiryDate>0920</expiryDate>

</ccInfo>

</creditCardDetails>

</creditCardData>

</mopInformation>

<dummy/>

</paymentModule>

</mopDescription>

</fopGroup>

</FOP\_CreateFormOfPayment>



Or

<PNR\_AddMultiElements xmlns="http://xml.amadeus.com/PNRADD\_14\_1\_1A">

<pnrActions>

<optionCode>0</optionCode>

</pnrActions>

<dataElementsMaster>

<marker1></marker1>

<dataElementsIndiv>

<elementManagementData>

<segmentName>FP</segmentName>

</elementManagementData>

<formOfPayment>

<fop>

<identification>CC</identification>

<creditCardCode>VI</creditCardCode>

<accountNumber>4444333322221111</accountNumber>

<expiryDate>0920</expiryDate>

</fop>

</formOfPayment>

<fopExtension>

<fopSequenceNumber>1</fopSequenceNumber>

<newFopsDetails>

<cvData>123</cvData>

</newFopsDetails>

</fopExtension>

</dataElementsIndiv>

</dataElementsMaster>

</PNR\_AddMultiElements>



After sending the request above, the FP elements is added to the booking.

/$--- MSC ---

RP/NCE1A0950/

RF AWSUI

1.FAMILYNAME/ADULTNAMEONE(ADT)(INFFAMILYINF/INFANT/03JAN14)

2.FAMILYNAME/ADULTNAMETWO(ADT)

3.FAMILYNAME/CHILDNAMEONE(CHD/01MAY10)

4 SU 262 E 03APR 5 LHRNRT DK3 1100 1035 04APR E 1 EQV MM

5 SU 261 E 15APR 3 NRTSVO DK3 1200 1610 15APR E 0 333 LS

SEE RTSVC

6 SU2584 E 15APR 3 SVOLHR DK3 1925 2125 15APR E 0 321 L

SEE RTSVC

7 AP NCE 33492943273 - AMADEUS PRODUCT PLANNING - A

8 TK OK29DEC/NCE1A0950

9 SSR INFT SU NN1 FAMILYINF/INFANT 03JAN14/S4/P1

10 SSR INFT SU NN1 FAMILYINF/INFANT 03JAN14/S5/P1

11 SSR INFT SU NN1 FAMILYINF/INFANT 03JAN14/S6/P1

12 SSR CHLD SU HK1 01MAY10/P3

**13 FP CCVIXXXXXXXXXXXX1111D0920\*CV**

## Fare\_PricePNRWithBookingClass

To price itineraries. Equivalent to FXX

<Fare\_PricePNRWithBookingClass>

<overrideInformation>

<attributeDetails>

<attributeType>**RP**</attributeType>

</attributeDetails>

<attributeDetails>

<attributeType>**RU**</attributeType>

</attributeDetails>

</overrideInformation>

<validatingCarrier>

<carrierInformation>

<carrierCode>**SU**</carrierCode>

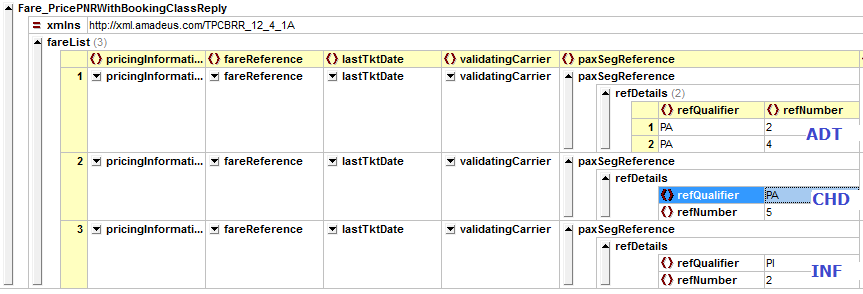
</carrierInformation>

</validatingCarrier>

</Fare\_PricePNRWithBookingClass>



There are 3 pricing solutions returned for each passenger type code (ADT, CHD and INF)



## Ticket\_CreateTSTFromPricing

To store the TST from the pricing

<Ticket\_CreateTSTFromPricing>

<psaList>

<itemReference>

<referenceType>**TST**</referenceType>

<uniqueReference>**1**</uniqueReference>

</itemReference>

</psaList>

<psaList>

<itemReference>

<referenceType>**TST**</referenceType>

<uniqueReference>**2**</uniqueReference>

</itemReference>

</psaList>

<psaList>

<itemReference>

<referenceType>**TST**</referenceType>

<uniqueReference>**3**</uniqueReference>

</itemReference>

</psaList>

</Ticket\_CreateTSTFromPricing>



There are 3 TSTs stored in this booking

/$T P/S NAME TOTAL FOP SEGMENTS

1 .1 FAMILYNAME/ADULTNAMEONE EUR 771.09 4-6

1 .2 FAMILYNAME/ADULTNAMETWO EUR 771.09 4-6

2 .3 FAMILYNAME/CHILDNAMEONE EUR 682.57 4-6

3 .1 I FAMILYINF/INFANT(INF) EUR 90.81 4-6

DELETED TST RECORDS MAY EXIST - PLEASE USE TTH

**/$TST00001** NCE1A0950 SL/07OCT I 0 LD 09OCT14 OD LONLON SI

T-

FXP

1.FAMILYNAME/ADULTNAMEONE(ADT)

2.FAMILYNAME/ADULTNAMETWO(ADT)

1 LHR SU 262 E 03APR 1100 OK EPX 03APR03APR 1PC

2 O NRT SU 261 E 15APR 1200 OK EPX 15APR15APR 1PC

3 X SVO SU 2584 E 15APR 1925 OK EPX 15APR15APR 1PC

LHR

FARE F GBP 250.00

EQUIV EUR 322.00

TX001 X EUR 270.00-YQAC TX002 X EUR 11.10-YRVB TX003 X EUR 91.29-GBAD

TX004 X EUR 57.81-UBAS TX005 X EUR 15.12-SWAE TX006 X EUR 3.77-OISE

TOTAL EUR 771.09 BSR 1.285658

GRAND TOTAL EUR 771.09

LON SU TYO202.71SU X/MOW SU LON202.71NUC405.42END ROE0.616616

\*SU \*

**/$TST00002** NCE1A0950 SL/07OCT I 0 LD 09OCT14 OD LONLON SI

T-

FXP

3.FAMILYNAME/CHILDNAMEONE(CHD/01MAY10)

1 LHR SU 262 E 03APR 1100 OK EPX CH25 03APR03APR 1PC

2 O NRT SU 261 E 15APR 1200 OK EPX CH25 15APR15APR 1PC

3 X SVO SU 2584 E 15APR 1925 OK EPX CH25 15APR15APR 1PC

LHR

FARE F GBP 187.00

EQUIV EUR 241.00

TX001 X EUR 270.00-YQAC TX002 X EUR 11.10-YRVB TX003 X EUR 91.29-GBAD

TX004 X EUR 57.81-UBAS TX005 X EUR 7.60-SWAE TX006 X EUR 3.77-OISE

TOTAL EUR 682.57 BSR 1.285658

GRAND TOTAL EUR 682.57

LON SU TYO152.03SU X/MOW SU LON152.03NUC304.06END ROE0.616616

\*SU \*

**/$TST00003** NCE1A0950 SL/07OCT I 0 LD 09OCT14 OD LONLON SI

T-

FXP

1.FAMILYINF/INFANT(INF)

1 LHR SU 262 E 03APR 1100 NS EPX IN90 03APR03APR 1PC

2 O NRT SU 261 E 15APR 1200 NS EPX IN90 15APR15APR 1PC

3 X SVO SU 2584 E 15APR 1925 NS EPX IN90 15APR15APR 1PC

LHR

FARE F GBP 25.00

EQUIV EUR 33.00

TX001 X EUR 57.81-UBAS

TOTAL EUR 90.81 BSR 1.285658

GRAND TOTAL EUR 90.81

LON SU TYO20.27SU X/MOW SU LON20.27NUC40.54END ROE0.616616

\*SU \*

## PNR\_AddMultiElement

To commit the PNR

<PNR\_AddMultiElements>

<pnrActions>

<optionCode>**10**</optionCode>

<optionCode>**30**</optionCode>

</pnrActions>

</PNR\_AddMultiElements>

<PNR\_Reply>

<pnrHeader>

<reservationInfo>

<reservation>

<companyId>**1A**</companyId>

<controlNumber>**3GYKOG**</controlNumber>

</reservation>

</reservationInfo>

</pnrHeader>

<securityInformation>

<responsibilityInformation>

<typeOfPnrElement>**RP**</typeOfPnrElement>

</responsibilityInformation>

</securityInformation>

<sbrPOSDetails>

<sbrUserIdentificationOwn>

<originIdentification>

<inHouseIdentification1></inHouseIdentification1>

</originIdentification>

</sbrUserIdentificationOwn>

</sbrPOSDetails>

<sbrCreationPosDetails>

<sbrUserIdentificationOwn>

<originIdentification>

<inHouseIdentification1></inHouseIdentification1>

</originIdentification>

</sbrUserIdentificationOwn>

</sbrCreationPosDetails>

<sbrUpdatorPosDetails>

<sbrUserIdentificationOwn>

<originIdentification>

<inHouseIdentification1></inHouseIdentification1>

</originIdentification>

</sbrUserIdentificationOwn>

</sbrUpdatorPosDetails>

<originDestinationDetails>

<originDestination></originDestination>

</originDestinationDetails>

<dataElementsMaster>

<marker2></marker2>

</dataElementsMaster>

</PNR\_Reply>



/$--- TST RLR ---

RP/NCE1A0950/NCE1A0950 SU/BB 7OCT14/0656Z **3GYKOG**

NCE1A0950/1749SL/7OCT14

1.FAMILYNAME/ADULTNAMEONE(ADT)(INFFAMILYINF/INFANT/03JAN14)

2.FAMILYNAME/ADULTNAMETWO(ADT)

3.FAMILYNAME/CHILDNAMEONE(CHD/01MAY10)

4 SU 262 E 03APR 5 LHRNRT HK3 1100 1035 04APR E SU/DOPEDR

5 SU 261 E 15APR 3 NRTSVO HK3 1200 1610 15APR E SU/DOPEDR

6 SU2584 E 15APR 3 SVOLHR HK3 1925 2125 15APR E SU/DOPEDR

7 AP NCE 33492943273 - AMADEUS PRODUCT PLANNING - A

8 TK OK07OCT/NCE1A0950

9 SSR INFT SU KK1 FAMILYINF/INFANT 03JAN14/S4/P1

10 SSR INFT SU KK1 FAMILYINF/INFANT 03JAN14/S5/P1

11 SSR INFT SU KK1 FAMILYINF/INFANT 03JAN14/S6/P1

12 SSR CHLD SU HK1 01MAY10/P3

13 FV PAX SU/S4-6/P1-2

14 FV PAX SU/S4-6/P3

15 FV INF SU/S4-6/P1

## Tips

* Dedicated / stateful sessions should be used for this flow
* To avoid fare discrepancies during the pricing and booking process
  + The same pricing options must be used in MPTB and Fare\_PricePNRWithBookingClass
  + The validating carrier returned by MPTB must be present in the PNR before pricing or it must be specified in the Fare\_PricePNRWithBookingClass request. This also ensures that the booking is eligible for ticketing
* To commit the PNR, option code 10 (ET) in PNR\_AddMultiElement is recommended to be used for online environment. This is to prevent simultaneous updates with the Airlines (to avoid the risk of not getting the Airline record locator returned)
* To prevent a PNR being saved when there is a warning message returned, the option code 30 in PNR\_AddMultiElement should be added. Without Option Code 30, a PNR will be saved and bypass the warning message.

## Error Handling

* In the event that an error occurs during the flow, the application should handle it by sending a Sign-out request to close the dedicated session and record the sessionID, timestamp and the query & reply for further investigation

# Certification Checklist

|  |  |
| --- | --- |
| **Certification check list** |  |
| **General** | **Incident Type** |
| The application must not create infinite loop(s) in the Amadeus host. | Error |
| Error handling must be properly implemented | Error |
| Application should properly read the XML response | Error |
| Data formats and values must be checked before being sent to Amadeus. In particular, date handling must be correct : the application must detect invalid or out of sequence dates and return an error to the user without sending a request to Amadeus | Error |
| Unnecessary or redundant transactions must be avoided. | Warning / Error |
| Structured messages must be used when available. | Warning |
| Duplicated booking must not be created. On web-based applications, it must be impossible to go back from the booking confirmation page and book the same itinerary again | Error |
| **Session management** |  |
| All sessions must be properly closed by the application. The Inactivity Time Out must not systematically be reached.  A keep-alive mechanism can be authorised in certain cases. This must be discussed between all parties during the implementation. | Error |
| Timers (inactivity/response) must be properly managed by the client application. | Error |
| The sequence number must be incremented on the client application side (Valid for Soap Header 1.0, 1.1, 2.0, 2.1) | Error |
| For public online applications, the session must be released while an action is expected from the end user | Warning |
| For SOAP Header version less than 4.0, a pool of sessions must be used for look/search transactions. | Error |
| Session header 4.0 should be implemented | Error |
| The Duty Code must be populated in the authentication message (Security\_Authenticate) for Soap Headers prior to 4.0 and in the Soap header itself for Soap header 4.0 and beyond | Warning |
| The application must be able to correctly log errors for incident reporting with Amadeus - for example, the error message, the session ID, and the Time Stamp must be logged. | Error |
| **Fare\_MasterPricerTravelBoard & Fare\_MasterPricerCalendar** |  |
| The application must incorporate the best practices when implementing Master Pricer products. | Error |
| For Session Headers 1.0, 1.1, 2.0, 3.0, Master Pricer Travel Board and Calendar requests must be sent from a pool of sessions. For session header 4.0, use a stateless session to send the Master Pricer Travel Board and Calendar requests. | Error |
| Master Pricer Calendar should be followed up by a Master Pricer Travel Board request to get a list of lowest available recommendations (up to 250) for the selected travel dates to be presented to the end user. | Error |
| Recommendations (fares and flights) returned by Master Pricer Travel Board and Calendar must be correctly displayed by the application | Error |
| Availability and price quotation should not be used as follow up transactions to Master Pricer Travel Board and Calendar | Error |
| The Office Id used for the Master Pricer Travel Board and Calendar requests must be the same as the booking OfficeID. | Error |
| The first validating carrier returned by Master Pricer Travel Board must be used for the subsequent booking, pricing and ticketing of the recommendation. | Error |
| The application must correctly handle passenger types (PTC) in the Master Pricer Travel Board and Calendar request. The same PTCs used in the Master Pricer request must be used in the pricing request. | Error |
| The application must be able to handle the pricing and ticketing of multiple passenger type codes (PTC). | Error |
| When using the Cabin option, it should never be used to specify Cabin Option= Economy (Y). Specifying Cabin Option=Y may remove the lowest fare from the results when the lowest fare is not fully economy; for example restricted business class can in fact be cheaper than full economy or the cheapest solution could be a mixture of Business and Economy. Additionally, depending on the departure dates, Economy classes may not be available and therefore no solution would be found by Master Pricer when Cabin Option=Y is requested. | Warning |
| Non Homogeneous recommendations (if requested) must be handled properly (booking, pricing, and ticketing) | Error |
| Multi-ticket recommendations (if requested) must be handled properly (booking, pricing and ticketing) | Error |
| If Corporate Fares are expected, the corporate code must be included in the requests | Error |
| When booking a recommendation corresponding to non-public fares (unifare, corporate or web fare), the pricing should be done on the non-public fare only. | Error |
| **Fare\_InformativePricing** |  |
| The Informative Pricing function must be used to verify pricing prior to end-user confirmation of the reservation | Comment |
| Pricing by Fare Basis with automatic validation of the rules (equivalent to FXX/A-) should be used when trying to price a specific fare after a fare search. | Warning |
| Connecting flights must be properly identified:  (<flightTypeDetails>  <flightIndicator>X</flightIndicator> </flightTypeDetails> <itemNumber>X</itemNumber>) | Error |
| **Air\_SellFromRecommendation** |  |
| The correct number of seats must be booked. No seat should be booked for infants defined as INF. Infants taking a seat should be defined as INS. | Error |
| Air segments must not be sold to check the availability or for pricing purposes. | Error |
| When algorithm M2 is used. The PNR must be ignored after an unsuccessful Air\_SellFromRecommendation (partially unsuccessful or not) | Error |
| Air\_SellFromRecommendation should be used to sell the recommendations returned by Master Pricer Travel Board. | Error |
| **All PNR** |  |
| To end-transact a PNR, the following elements are mandatory :  Name (NM) Contact (AP, APE, AM,…) Itinerary segment (air, hotel, car,…) Ticketing Agreement (TK)  Received From (RF) When retrieving and updating an existing PNR, a new RF element must be entered prior to EOT (unless the office profile is set to insert a default RF at EOT). | Error |
| The PNR must be created in the Amadeus Host only when user confirms the reservation. | Error |
| Passengers must be correctly defined in the PNR. | Error |
| Infants must be associated to adults. | Error |
| Infant age or Date of Birth must be specified in the name of the INF for correct automatic creation of the SSR INFT. | Error |
| For public online applications, if a PNR is end-transacted without form of payment element (FP), a TKXL element must be present. Creating a TKOK element without a FP element may lead to the cancellation of the seats by automated airline processes. | Error |
| Additional passengers should not be added to an existing, saved PNR. Instead, an additional PNR should be created. | Error |
| If the traveller is going to certain countries, it is now mandatory to add **APIS information** to be processed by the authorities of these countries. USA, Mexico and Spain are among countries that require APIS information. | Comment |
| The only characters allowed in a passenger’s name are A to Z and SPACE. Your application should either disallow other characters, or convert them to a SPACE. | Error |
| Airlines’ definition of an Infant passenger is 0 to under 2 years old at time of commencement of the last segment of the journey. A Child is generally regarded as being between 2 and under 12 years old. Your application should validate the ages of these passengers against the dates of travel when these passenger type codes are specified. | Error |
| If the pricing and/or TST creation is in the same flow as the segment sell, it is better to save the PNR only at the end of the complete flow. Otherwise, if there is an issue with the pricing, the PNR will have already been saved and you may have to either cancel the PNR or queue it for manual processing. | Error |
| If using PNR\_AddMultiElements with Option code 11 (End Transaction and Redisplay), Option code 267 must also be used (stop at the 1st sell error). This ensures air sell errors are detected and avoids unwanted end of transaction. | Warning |
| No more than 9 passengers must be treated in a single booking. If more passengers are required, the group booking feature must be used. | Error |
| Option Code 30 must be used in the PNR\_AddMultiElements when End Transacting the PNR to get end of transaction warnings. A Warning message will not prevent a PNR being saved, but an additional End Transaction command (without Option Code 30) will have to be sent in order to save the PNR and effectively bypass the Warning message. | Comment |

# Certification Test Cases

* **Master Pricer Search**

|  |  |
| --- | --- |
| **Scenario** | Description |
| Search 1ADT + business class | This is to validate the filing of the Master Pricer query when 1 ADT and the cabin class are requested, and to make sure the recommendations display properly |
| Search 2ADT, 1CH and 1INF | This is to validate the proper use of the Master Pricer query when 2ADT, 1INF, and 1 CHD are requested. |

* **Error Management for Master Pricer Search**

|  |  |
| --- | --- |
| **Scenario** | Description |
| Unchronological Date Search | Master pricer cannot return flights for dates that are not in chronological order. This test case is to validate that searches requesting unchronological dates are rejected before being sent to Amadeus. This will save time, as the customer will be prompted to correct their mistake immediately, as well as avoid sending unnecessary transactions to Amadeus. |
| Past Date Search | Master pricer cannot return flights for past dates. This test case is to validate that searches requesting past dates are rejected before being sent to Amadeus. This will save time, as the customer will be prompted to correct their mistake immediately, as well as avoid sending unnecessary transactions to Amadeus. |
| Future Date Search | Master Pricer can only return journeys with travel dates up to 361 days into the future. This test case is to verify that future dates greater than 361 days are not being sent to Amadeus. |
| Too many Infants | Infant passengers are generally sold without a seat and must be associated to an adult passenger. As there can only be as many infants without a seat as the number of seats being sold, it is important that a check is carried out before sending a query to Amadeus to check that the number of infants does not exceed the number of seats. |
| Too many Passengers | A maximum of nine passengers can be in a Master Pricer request. This test case is to validate that a check has been put into place to inhibit the sending of searches for more than nine passengers. |

* **Fare Informative Pricing transaction**

|  |  |
| --- | --- |
| **Scenario** | Description |
| Get the fare information of a selected Master Pricer recommendation for a round trip for 2ADT, 1CH and 1INF | This is to validate that the application is able to create a query correctly; passenger type codes, pricing options, validating carrier and flight details |
| Get the fare information of a roundtrip with connecting flights. | This is to validate that the application is able to handle connecting flights properly |

* **Booking**

|  |  |
| --- | --- |
| **Scenario** | Description |
| Book a round trip for 1ADT + business class | This is to validate that the application can book a recommendation returned by Master Pricer for 1 adult on business class |
| Book a round trip for 2ADT, 1CH and 1INF | This is to validate that the application can book multiple types of passengers Description |

* **Error Management for Booking**

|  |  |
| --- | --- |
| **Scenario** | Description |
| Book a round trip for 2ADT on a class that is not available | This is to validate the application’s error detection process; the application should be able to handle errors during the booking process. |
| Book 1 INF – Return Trip | This is to validate the application’s error detection process; the application should not allow infants to be booked alone |
| Book 1 ADT with name : “Test1” | This is to validate the application’s error detection process; the application should not allow any invalid characters to be entered in the name element |
| Book 1 ADT + 1 INF with DOB = 01/01/01 | This is to validate the application’s error detection process; the application should not allow invalid DOB to be entered for an infant |
| Create a PNR then click on the Back Button | This is to validate the Back button usage after the booking is completed; the application should not book 2 PNRs and the session should be closed once the booking is completed |

# References

**Product overview for Shopping product:** <https://webservices.amadeus.com/catalogue/viewCatalogue.do?id=277&type=1>

**User guide for each service:** <https://webservices.amadeus.com/extranet/functional_documentation.do>

**Session Management:**

Soap2 : <https://webservices.amadeus.com/extranet/best_practices_session_management.do>

Technical document <https://webservices.amadeus.com/extranet/kdbViewDocument.do?externalId=wikidoc_wbs_frameworks&docStatus=Published&mpId=fla__1__technical>

**Certification Checklist:** Amadeus Web Services Certification Checklist Master Document.xlsx, if this document has not yet been provided to you, please check with Web services Implementation Manager.